Using Deep Learning for Detecting Spoofing Attacks on Speech Signals

Alan Godoy[1,2], Flávio Simões[1], José Augusto Stuchi[1], Marcus de Assis Angeloni[1],
Mário Uliani[1], Ricardo Violato[1]

[1]CPqD Foundation, Campinas, Brazil
[2]University of Campinas, Campinas, Brazil

{amello,simoes,jastuchi,massis,uliani,rviolato}@cpqd.com.br

Abstract

It is well known that speaker verification systems are subject to spoofing attacks. The Automatic Speaker Verification Spoofing and Countermeasures Challenge – ASVSpoof2015 – provides a standard spoofing database, containing attacks based on synthetic speech, along with a protocol for experiments. This paper describes CPqD's systems submitted to the ASVSpoof2015 Challenge, based on deep neural networks, working both as a classifier and as a feature extraction module for a GMM and a SVM classifier. Results show the validity of this approach, achieving less than 0.5% EER for known attacks.

Index Terms: Speaker Verification, Spoofing Countermeasures, Deep Neural Networks

1. Introduction

Biometric spoofing is usually described as a direct attack perpetrated against a biometric authentication system by presenting it a fake (forged or copied) biometric sample. Anti-spoofing refers, therefore, to countermeasures designed to detect and prevent these attacks [1].

In the last few years, many studies have shown that even state-of-the-art automatic speaker verification (ASV) systems are vulnerable to such attacks, which can be based on a variety of techniques, including voice conversion, speech synthesis, artificial signals, impersonation, and replay [1]. Although most of these studies proposes countermeasures too, they usually are based on prior knowledge about the attack method, what is clearly unrepresentative of real world scenarios. Additionally, each one is also based on its own database, protocol and metrics, making it difficult to perform a proper analysis of results and restricting fair comparison among them [2].

The recent Automatic Speaker Verification Spoofing and Countermeasures Challenge, ASVspoof2015[1], which focused on spoofing attacks based on synthetic speech, provided the first standard spoofing database along with a protocol for experiments. Differently from previous works, 10 different voice conversion and speech synthesis algorithms were used to generate the database, but only 5 of them were known in advance in order to train spoofing detection algorithms [3]. This paper describes the systems based on neural networks submitted to the challenge and analyze the obtained results.

Deep Neural Networks (DNN) have been widely used in a variety of research fields, such as image classification [4, 5], natural language processing [6] and information retrieval [7]. In the speech processing community, DNN have been applied to speech recognition [8], speech synthesis [9, 10] and also to speaker recognition [11, 12].

One straightforward application of a DNN for spoofing detection is to use it as a classifier, whose input data can be either raw audio [13] or features previously extracted from the audio files. A natural choice for audio pre-processing is to use features proven to yield good results in speaker recognition and spoofing detection tasks, such as traditional Mel Frequency Cepstral Coefficients (MFCC) [14] and Modified Group Delay Cepstral Coefficients (MGDCC) [15], which have been broadly used not only in combination with neural networks, but also with a handful of other classification algorithms.

In problems like spoofing detection, a DNN can also be employed as a feature extraction module itself, by means of a bottleneck approach [16]. In this case, a network, initially trained for regression or classification, has its final layers removed, and the output of its last remaining layer is used as a new representation of the input data for future classification [13]. The network can receive as input a pre-processed feature vector, a high-level full representation of the signal (using, for instance, the Fast Fourier transform) or even the raw audio. In this work, we used the high-level representation approach, as described in Section 3.

The paper is organized as follows: Section 2 presents a brief description of neural networks. Section 3 explains the methods applied. Section 4 presents and discusses results obtained on the ASVspoof2015 challenge. Finally, Section 5 draw some conclusions, as well as points to topics for future research.

2. Neural Networks

The submitted systems are based on a Deep Learning approach. A deep neural network (DNN) is an artificial neural network with more than one hidden neuronal layer between its inputs and outputs [17]. The DNN concept can be implemented using many different architectures, such as Convolutional Neural Networks (CNN) [18], Autoencoders [19], and Multilayer Perceptrons (MLP) [20].

In a Multilayer Perceptron, tipically, each neuron j in a hidden layer l employs a sigmoid function, such as the logistic function or hyperbolic tangent, to map the total input x_j^l, received from the layer $l-1$, to an output y_j^l, that is sent to the following layer, $l+1$.

$$x_j^l = b_j^l + \sum_{1 \le i \le N^{l-1}} w_{i,j}^l y_i^{l-1} \qquad (1)$$

$$y_j^l = logistic(x_j^l) \qquad (2)$$

where N^{l-1} is the number of neurons in layer $l-1$, y_i^{l-1} is the output of neuron i on previous layer, $w_{i,j}^l$ is the connection

---

[1]http://www.spoofingchallenge.org/

weight between neuron i from layer $l-1$ and neuron j from layer l, and b_j^l is the bias of neuron j of the current layer [17].

One of the major DNN applications is for multiclass classification problems. In this context, a softmax nonlinear function can be used in the network output layer to convert inputs x_j^{out}, into a class probability, p_j:

$$p_j = \frac{\exp(x_j^{out})}{\sum_{1 \le k \le N^{out}} \exp(x_k^{out})} \quad (3)$$

where N^{out} is the number of neurons in the output layer, which is equal to the number of possible classes. In this case, the network output p_j will indicate the likelihood of the input fed to the network belonging to the j-th class [17].

3. Method

3.1. Feature Extraction

Aiming at detecting if an audio is authentic or not, a deep neural network based on a multilayer perceptron architecture was used as a feature extraction module. In a bottleneck approach, the network output layer is removed and the activations of the last hidden layer neurons are treated as new features for future classification. Figure 1 shows how audio was processed, from feature extraction to network supervised training.

Instead of feeding raw signal directly as input to the network, a pre-processing step was performed in order to transform input signals into sequences of feature vectors. This decision was based on preliminary tests, which indicated such a step was able to improve the learning rate and allowed the use of more compact networks. Therefore, each signal file is divided into a sequence of 20 ms consecutive non-overlapping frames. No window function is applied. In parallel, a voice activity detection method based on ITU G.729B [21] is applied, so each frame is classified as speech/non-speech and only speech frames are preserved.

Different representations were tested as input for the MLP, including the raw speech frame itself, MFCC, MGDCC and Discrete Fourier Transform (DFT) coefficients. Nevertheless, better results were achieved with the Discrete Cosine Transform (DCT) coefficients. The DCT has the energy compaction property, which concentrates most of the signal information in a few low-frequency components [22]. For this reason, the first 128 DCT coefficients are used as feature for each active speech frame.

In order to avoid loss of long term information that can possibly be used to distinguish spoofing attacks, when an input is presented to the MLP, each central speech frame is surrounded by its ten previous frames and the ten following ones, including silence frames [11]. Thus, a vector with 2688 features is used as network input.

The backpropagation algorithm, in conjunction with the Stochastic Gradient Descent optimization technique [20], was applied to train the network to classify whether the input represents an authentic (human) or spoofed audio frame. Ground truth consists of a label indicating if the input audio is authentic or belongs to one of five spoofing categories, named S1, S2, S3, S4 or S5 [2].

Preliminary experiments indicated that using only two classes – spoofing and human – as output led to poor perfomance in class S1. One hypothesis is that this could happen because S1 distinguishes from other attacks since it is based on a unit selection algorithm, which concatenates pieces of authentic signal to create a new audio. To deal with this, it was
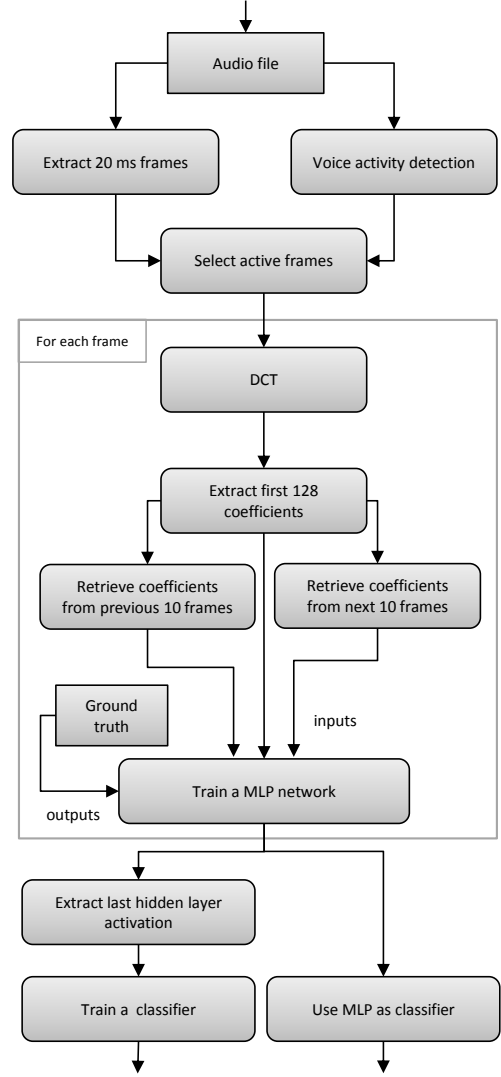

Figure 1: Basic flowchart used for spoofing detection

decided to drive the network training towards distinguishing S1 from the other spoofing attacks, increasing the relevance (on network performance) of detecting borders between pieces of authentic speech. Thus, three classes were created, as depicted in Table 1: authentic human speech (100), S1 spoofing attack (010) and other spoofing attacks (001).

Figure 2 shows the MLP deep architecture used in this paper. 1024 neurons were used in the first hidden layer, 512 in the second hidden layer and 32 in the last one. The last hidden layer is artificially small in order to create a bottleneck, which compress signal information useful for spoofing classification in a low-dimensional representation [16]. Each hidden layer uses the logistic function as activation. The output consists of 3 neurons, each one with softmax activation function, returning a real number between 0 and 1. After finishing the network training, the output layer was removed and the activations of the last hidden layer neurons were used as new output, extracting the bottleneck features, as indicated in Figure 2.

Table 1: *MLP classes output meanings.*

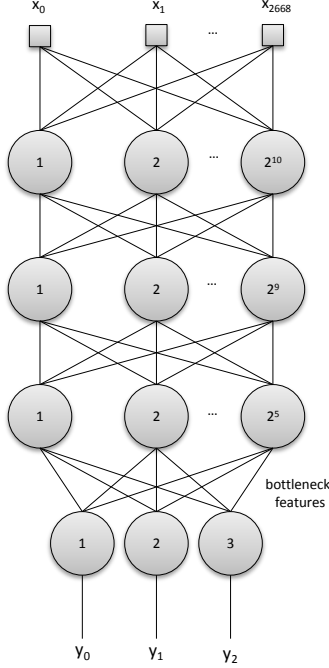| y0	y1	y2	Meaning
1	0	0	human
0	1	0	S1 attack
0	0	1	S2, S3, S4, S5 attacks

Figure 2: MLP used for feature extraction and classification

3.2. Classification

Three different classifiers were tested: Support Vector Machines (SVM), Gaussian Mixture Models (GMM) and Multilayer Perceptron. In the cases of the SVM and the GMM classifiers, feature extraction took an additional step. Since each audio file has a different duration and, thus, a different number of frames, feature vectors over all frames were averaged so that each file was represented by a single fixed-size 32-dimensional feature vector [23].

A SVM classifier [24] based on the Radial Basis Function (RBF) kernel was generated. Samples from the training set were computed and used to train the SVM-RBF. All spoofing attacks were considered as a single negative class for training.

The SVM-RBF classifier parameters C (controls the cost of misclassification on the training data) and γ (parameter of a Gaussian kernel to handle nonlinear classification) were tuned by performing grid search with K-fold cross-validation over the train set, using 5 folds. Values of 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0 and 10000.0 were searched both for C and γ. Optimum parameters were chosen aiming at minimizing the average equal error rate (EER) over all 5 folds. After this search, optimum values of $C = 0.1$ and $\gamma = 10$ were found and the SVM-RBF classifier was retrained with the whole training set. SVM-RBF outputs vary in the interval $[0.0, 1.0]$ and represent the likelihood of the test sample belonging to positive class, i.e., authentic speech audio.

For the GMM based classifier, two GMMs were trained, one with authentic audios and another with spoofed audios. The following number of Gaussian mixtures were tested: 4, 8, 32, 64, 128, 256 and 512, wherein 8 mixtures gave the lowest EER on the development set. The classifier output is given by the log-likelihood ratio of authentic GMM with respect to spoofing GMM.

Figure 3 shows the log-likelihood ratio (score) distribution obtained on the development set when a 8-mixture GMM was employed to classify the bottleneck features. Score values vary in the interval $[-\infty, +\infty]$ and the higher the value, the higher the probability of the tested sample being authentic. The figure clearly shows this strategy provided a good separation over the develpment set. A similar behavior was verified for the SVM-RBF classifier.
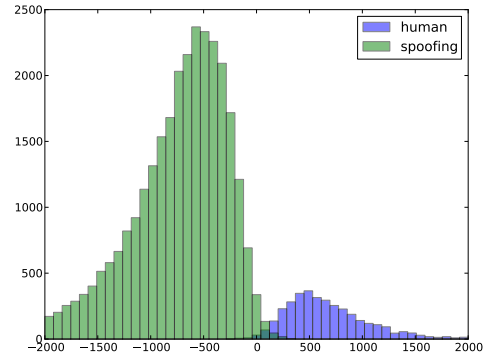

Figure 3: Scores distribution for spoofing (green) and authentic (blue) audios on the development set when using a GMM with 8 gaussians and bottleneck features

The third and last tested approach consisted of using the MLP trained for feature extraction directly as a classifier, without the removal of the output layer. In this case, the feature extraction was merged with the classification step.

As the network last layer returns three values using the softmax function, according to presented in Figure 2, only y_0 is considered, since it represents the likelihood of being an authentic speech. Thus, values for this third approach vary in the interval $[0.0, 1.0]$. A score (y_0) was then calculated for each frame in the audio file, generating a score array for the entire audio. This array was used to compute a unique score for the audio sample. To do so, aiming at removing outliers within the audio file, the first 15% lower array values are removed as well as the 25% higher values. The remaining 60% of the scores were then averaged, resulting in the final score.

These three aproaches were, then, applied to the evaluation set, which contained samples comprising both known and unknown attacks. Results are presented in the next section.

4. Results and Analysis

Results obtained for the three tested systems are summarized in Table 2. According to challenge rules, the adopted metric is the EER. For more details on what that means and how it is calculated, please refer to the contest evaluation plan [3].

It can be seen that:

- the SVM-RBF classifier showed the best performance

Table 2: *EER results (%) obtained on development set and on evaluation set for known and unknown attacks.*

Classifier	Dev Set	Known	Unknown	All
SVM	0.491	0.412	13.026	6.719
GMM	0.658	0.443	12.796	6.620
MLP	0.631	0.464	12.589	6.527

for known attacks, while the unknown attacks were better detected by the MLP classifier. However, EER values are very close, which means that the choice of the classifier is less determinant for the overall performance than the feature extraction mechanism itself.

- all three systems performed very well for the known attacks, which shows that the network was successfull in capturing the pattern of attacks learned during training.

- most of the unknown attacks were correctly detected; however a clear degradation of performance can be observed when error rates of known and unknown attacks are contrasted.

- when considering only the five unknown attacks discriminated by method used (these results are not shown here due to space reasons), the proposed method obtained good results (EER near to 1%) in three of them. Results for attacks S8 (a tensor-based voice conversion) and S10 (a speech synthesis algorithm implemented using the open source MaryTTS system), however, indicate a poor performance, with EERs of 26.8% and 31.7%, respectively.

One hypothesis for the degradation observed in classifiers' performances for evaluation set is the occurence of overfitting to noise present in training samples. This situation can be verified by the existence of a significant difference in error rates even when training and testing samples are drawn from the same distribution. That is not what the results presented here show, since performance in the development set is close to the performance for known attacks in evaluation set.

The second hypothesis is lack of generalization capacity, which means that some of the distinctive features learned by the network and the classifiers are not related to what distinguishes an authentic recording from spoofing attacks in general, but are rather due to patterns only observed in the known attack samples, i.e., specific characteristics of synthesis and conversion algorithms used during training step.

It was also verified after the submission that many spoofing audios available on the training and development sets present descontinuity in low frequency noise, mainly in the range 0 to 100 Hz. Figure 4 shows the problem. In this case, as 128 DCT coefficients was used as DNN input, the first coefficients will indicate this discontinuity and the network will learn this characteristic as a relevant feature to distinguish authentic from spoofing audios, degrading the network's generalization capacity when audios without this discontinuity are presented.

Even though some degradation of performance is expected, the results obtained show that there is room for improvements, since the nature of unknown attacks is not inherently different from that of the known ones.

5. Conclusions

The study presented here comprises the results obtained, along with the description of the systems implemented by CPqD for
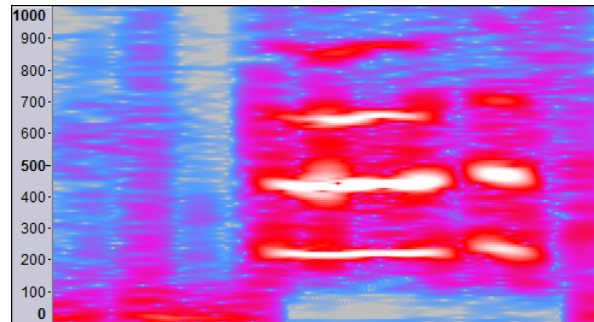

Figure 4: Low frequency noise discontinuity available on training and development set (0 to 1000 Hz in vertical axis)

the Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVSpoof2015), held as a special session in INTERSPEECH 2015. The main goal of the challenge was the detection of spoofing attacks based on sinthesized and transformed speech.

A speech feature extraction framework based on deep neural networks for spoofing detection is presented. The network can be used as a classifier itself or can be viewed as a bottleneck feature extractor feeding other classifiers. Two different classifiers were tested: a Gaussian Mixture Model and a Support Vector Machine with the radial basis function.

The proposed systems were trained with the training set and tested on two different evaluation sets: one with attacks similar to those presented during training and another with unknown attacks, just as described in the evaluation plan.

The use of a DNN as a feature extractor is of particular interest, as the generated features are fine-tuned to provide a good representation specifically for the problem to be solved, be it spoofing detection, speaker/speech recognition or other tasks. However, these features are highly dependent on the training samples and they can learn any bias present in this set. Thus the careful design of large and diverse datasets is even more relevant when using this kind of feature.

Performance for the known attacks was satisfactory ($EER < 0.5\%$), indicating the adequacy of the proposed strategies. Results obtained for the unknown attacks were also promising. For some of the new attacks, however, the detection strategy had poor performance. This could be easily overcome with training data composed by samples generated by a more diverse attack techniques. In addition to an improved training set, the use of alternative forms of parametrization of the input audio in the neural network could be beneficial. Representations that make the speech phase spectrum more evident are specially interesting, as the use of such information proved to be highly successful in literature for spoofing detection [15].

Lastly, in future work, other network architectures, like Convolutional Neural Networks, should be tested in order to study which of them is able to provide better detection of unknown attacks, an ability extremely relevant in real-world applications, as rarely the techniques used by fraudsters for identity theft are known in advance.

6. Acknowledgements

7. References

[1] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and counter-measures for automatic speaker verification." in *INTERSPEECH*, 2013, pp. 925–929.

[2] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association, September 6–10, Dresden, Germany, Proceedings*, 2015, p. Submitted.

[3] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "Asvspoof 2015: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," *Training*, vol. 10, no. 15, p. 3750, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[5] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[6] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *International Conference on Machine Learning (ICML)*, 2008, pp. 160–167.

[7] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using click-through data," in *Association for Computing Machinery (ACM) International Conference Information and Knowledge Management (CIKM)*, 2013.

[8] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," vol. 20, no. 1, pp. 30–42, 2012.

[9] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8012–8016.

[10] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.

[11] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[12] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[13] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[14] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[15] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7234–7238.

[16] D. Yu and M. L. Seltzer, "Improved bottleneck features using pre-trained deep neural networks." in *INTERSPEECH*, vol. 237, 2011, p. 240.

[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[18] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition." in *INTERSPEECH*, 2013, pp. 3366–3370.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1953039

[20] S. O. Haykin, *Neural networks and learning machines*. Pearson Education Upper Saddle River, 2009, vol. 3.

[21] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit, "Itu-t recommendation g. 729 annex b: a silence compression scheme for use with g. 729 optimized for v. 70 digital simultaneous voice and data applications," *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 64–73, 1997.

[22] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *Computers, IEEE Transactions on*, vol. 100, no. 1, pp. 90–93, 1974.

[23] M. J. R. F. Correia, "Anti-spoofing: Speaker verification vs. voice conversion," Master's thesis, Instituto Superior Técnico Lisboa, 2014.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018